

**5.1. Haz un proyecto desde cero que contenga un formulario con dos campos de texto (nombre y edad) y un sencillo tratamiento desde un controlador (puedes tomar el código de la captura de imagen del principio de este capítulo en los apuntes, que hace exactamente lo que se solicita).**

//Hay proyecto

**5.2. Haz una copia del proyecto 4.3. y vamos a cambiar la forma de comunicaros con la aplicación.**

Vamos a sustituir las URL con variables en la parte query y en la parte path por formularios. Veremos como la capa de servicio permanece intacta (así comprobamos las ventajas de la separación por capas).

Lo haremos paso a paso:

- a) Haz una página de inicio que simplemente contienen dos enlaces, uno para una página de cálculos numéricos y otro para una página de tratamiento de fechas.
- b) La página de cálculos numéricos contendrá un formulario con un solo campo de texto en el que se introducirá un número. Al pulsar el botón de envío devolverá una vista indicando si el número es primo o no. Una vez que te funcione puedes, mediante las anotaciones de validación, controlar si se introduce el campo de formulario vacío o con valores no numéricos.
- c) Añade otro formulario a la misma página, con dos campos de texto en donde introduciremos los catetos de un triángulo y un nuevo botón de envío. Al pulsarlo nos devolverá una vista con el valor de la hipotenusa.
- d) Añade a la misma página un tercer formulario similar a los anteriores que muestre los divisores del número introducido. Puedes usar el mismo CommandObject que en el apartado b).
- e) Vamos ahora con la página de tratamiento de fechas. Deberá contener un formulario que contenga dos cajas de texto y dos botones de radio agrupados. El primer botón de radio es para “días entre fechas” y el segundo para “años bisiestos entre fechas”. Si se selecciona el primero, al pulsar el botón de envío mostrará los días comprendidos entre las fechas introducidas en los campos de texto y si selecciona el segundo botón de radio, al pulsar el botón de envío, mostrará la cantidad de años bisiestos comprendidos entre esas fechas. Habrá que verificar que las fechas son válidas, en caso contrario redirigirá a una página de error.

//Hay proyecto

**5.3. Volviendo al proyecto del ejercicio 3.3 (la tienda), en la página contacto, añade un formulario con cinco campos: un texto para el nombre, un texto que debe ser de tipo email, una lista desplegable con los valores (queja, consulta, otros), una caja de texto para comentarios y un check de “Aceptar las condiciones del servicio”. Al enviarlo comunicar al usuario que ha sido recibido correctamente y mostrarle los datos que envió.**

Hay proyecto

**5.4. Realiza una aplicación con un formulario como el de la figura, en la que, al seleccionar un país en el desplegable, muestre su capital y población. Para ello habrá que realizar los siguientes pasos:**

La clase de servicio deberá incluir una colección con los datos de los países (nombre, capital, población), por ejemplo:

```
private List<Pais> paises = new ArrayList<>();
```

y los métodos necesarios para su gestión:

- cargarPaisesDesdeFichero(): empleará el método estático Files.readAllLines() para pasar todas las líneas del archivo países2019.csv proporcionado por el profesor a un List<String>.

Luego, aplicando el método split() a cada String podremos cargar la colección definida en el

punto anterior. Este método se debe ejecutar al iniciar la aplicación, por lo tanto, se incluirá en un CommandLineRunner.

- getPaises(): Para pasarle al controlador (y de ahí al desplegable de la vista) todos los nombres de todos los países.

- getPais (String nombre) : Obtendrá los datos de un país concreto a partir de su nombre.

Se invocará desde el controlador cuando se envíe el formulario, con un país seleccionado.

Este formulario es distinto a los vistos previamente ya que la página ruta destino del submit es la

misma que la de presentación del formulario. Esto hará que, en el controlador, el GET y POST atiendan a la misma ruta.

Finalmente, para hacer más ágil la aplicación, se puede eliminar el botón de submit del formulario y

añadir el código JavaScript necesario para que, al cambiar un valor de la lista desplegable, se envíe el formulario: <select onchange="this.form.submit()" th:field="\*{nombre}">

El formulario solo tiene un solo atributo, el nombre del país, deberíamos hacer un objeto(Command Object) con solo ese atributo pero, por comodidad, podemos hacer que el Command Object sea de tipo 'País'.

//Este lo tengo mal, no lo tengo para que lea los países desde un fichero txt, ya que en los recursos para ejercicios que descargara, aún no estaba ese fichero, y no miré si lo actualizara hasta hoy.

**5.5. (Opcional) Añade al formulario del ejercicio 5.3 la posibilidad de enviar un fichero adjunto en el formulario de contacto. El fichero se guardará en el servidor en la ruta /uploadFiles.**

//Este y 2 del siguiente ejercicio no fuí capaz de hacérselos(Sí, porque me puse muy tarde con ellos y no pude preguntarle.)

**5.6. Toma el proyecto MasterMind que te entregado por el profesor y realiza las siguientes tareas:**

**a) Estudia el código y comenta su estructura y funcionamiento.**

Primero hay un documento HTML que te muestra las normas del juego. Tras ello, cuando presiones el enlace, se calcula el número que debes adivinar(que se escribe por la pantalla del VSc, por si quieres comprobar casos concretos de cómo funciona(acertar o fallar X números, poniéndolos en posiciones incorrectas...)

**b) ¿Para qué incluye en el controlador y servicio la anotación @Scope("session") ?**

Para que si lo abres en varios navegadores puedas jugar entre ellos, y comparten la combinación numérica a buscar, y también los fallos. Se van viendo los fallos reales en cuanto actualizas.

**c) Modifica el proyecto de forma que en la página de inicio se solicite mediante un campo de texto la cantidad de intentos que tiene cada jugador para adivinar el número y mediante una lista desplegable la cantidad de dígitos que tendrá el número a adivinar.**

//No fui capaz de este apartado(De hecho cuando lo intenté me rompió una, y otra vez...), y me puse muy tarde para poder preguntarle.

**d) Modifica el proyecto para que en la página de juego muestre la cantidad de intentos restantes y también si se produce algún error cuando el usuario introduce un intento de adivinar el número (longitud incorrecta, número con duplicados, dígitos no numéricos o número introducido previamente).**

//Lo mismo